CSCI-B 565 DATA MINING
Homework 2
Morning Class
Computer Science Core
Spring 2015
Indiana University,
Bloomington, IN

Suhas Gulur Ramakrishna
suhgulur@indiana.edu

02/16/2015

All the work herein is solely mine.

Problems

Problem 1

a
Answer: The sample space Ω =
{ (1,1) (1,2) (1,3) (1,4) (1,5) (1,6)
(2,1) (2,2) (2,3) (2,4) (2,5) (2,6)
(3,1) (3,2) (3,3) (3,4) (3,5) (3,6)
(4,1) (4,2) (4,3) (4,4) (4,5) (4,6)
(5,1) (5,2) (5,3) (5,4) (5,5) (5,6)
(6,1) (6,2) (6,3) (6,4) (6,5) (6,6)}

b
Answer:

$$X = \Omega \rightarrow \Re = X(i+j) = \{2,3,4,5,6,7,8,9,10,11,12\}$$

c
Answer: g(X) =g(i,j) = { 9.99 if i + j = 9
- 1.00 if i + j ≠ 9 }

d
Answer: Mass function over X [1] =

X	2	3	4	5	6	7	8	9	10	11	12
P(X)	1/36	2/36	3/36	4/36	5/36	6/36	5/36	4/36	3/36	2/36	1/36

The graph is included in the image "PMF.jpg"

e
Answer: Mass function over g(X) =

g(X)	9.99	-1
p(g(X))	1/9	8/9

The graph is included in the image "PMFg(x).bmp"

f
Answer: E[X] = 2 * P(X=2) + 3 * P(X=3) + 4 * P(X=4) + 5 * P(X=5) + 6 * P(X=6) + 7 * P(X=7) + 8 * P(X=8) + 9 * P(X=9) + 10 * P(X=10) + 11 * P(X=11) + 12 * P(X=12)

E[X] = 2 * 1/36 + 3 * 2/36 + 4 * 3/36 + 5 * 4/36 + 6 * 5/36 + 7 * 6/36 + 8 * 5/36 + 9 * 4/36 + 10 * 3/36 + 11 * 2/36 + 12 * 1/36

E[X] = (2 + 6 + 12 + 20 + 30 + 42 + 40 + 36 + 30 + 22 + 12) / 36
E[X] = 252/36 = 7

g
Answer:

E[g(x)] = 9.99 * P(9.99) + (-1) * P(-1)
= 9.99 * 1/9 - 1 * 8/9
= 1.11 - 8/9
= 0.2211

g
Answer:

Since the probability of cookie(8/9) is more than the probability of coffee(1/9) the winner will be the guy with cookies.
In terms of profit
Coffee Guy= P((i+j) = 9) * 3 = 3/9 = 1/3 = 0.33
Cookie Guy= P((i+j) $\neq$ 9) * 1 = 8/9 * 1 = 0.89
The cookie guy will having more cookies compared tom coffee guy.

Problem 2

a
Answer: The potential uses of linear regression equations are [2]:

1) *Pure Description*: Regression is often used to determine how much specific factors such as the price of a commodity, interest rates, particular industries or sectors(independent variables) influence the price movement of an asset (dependent variable Y). It gives the good description of response variable by searching the equations with smallest residual sums of squares to make sure the data is fitting with equation with small residual sum pf squares.

2) *Prediction and Estimation*: Another major advantage of linear regression is to predict the future response for a given input.
The line of regression of Y on X is given by Y = a + bX where a and b are unknown constants known as intercept and slope of the equation. This is used to predict the unknown value of variable Y when value of variable X is known.

3) *Extrapolation*: Extrapolation is the process of taking data values at points x1, ..., xn, and approximating a value outside the range of the given points. This is most commonly experienced when an incoming signal is sampled periodically and that data is used to approximate the next data point. The data is modeled by a curve and linear regression is applied and sometimes

transformation also. The procedure followed is to first find the least-squares regression curve, and evaluate that function at that point. The error in our extrapolated value depends on how far we are from the mean of the x values.

4) *Estimation of parameters*: A good description of response variable, fit the data into least squares and search the equation with small residual sum of squares. The best solution would be to retain all parameters but in some cases little will be sacrificed if few variables are deleted. Mostly squared multiple correlation is used interpret the model(R squared).

5) *Control*: The concept of control is concerned with controlling the level of output by varying the levels of inputs. In this situation accurate estimates of of regression coefficients are desirable.
The easiest one can come up with, is to stratify the data so to have sub-groups with similar characteristics - there are then methods to pool those results together to get a single "answer". This works if there is a very small number of variables to control for, but typically, this rapidly falls apart as the data is split into smaller and smaller chunks.

6)*Model Building*: The model is the procedure of deleting extraneous variables or retaining relevant variables. The different procedures are

a) Considering all possible regressions

b) Stepwise method - Forward Selection or Backward Elimination

c) Optimal subsets

d) Sub-optimal methods

e) Ridge Regression

Ridge Regression : It is one of the type of choosing a model(by inherent deletion of variables) from a pool of variables. It is basically for problem involving non-orthogonal predictors.

Due to multicollinearity least squares estimates are unbiased but their variances are large so they may be far from the truth value.
In ridge regression, the first step is to standardize the variables (both dependent and independent) by subtracting their means and dividing by their standard deviations. In ordinary least squares, the regression coefficients are estimated using the formula.

$$\beta(K) = (X'X + K)^{-1}X'Y$$

K is a diagonal matrix whose components is to be determined . X is adjusted for of the data. t can be shown that there exists a value of k for which the mean squared error (the variance plus the bias squared) of the ridge estimator is less than that of the least squares estimator. The values for which $\beta(k)$ is small are deleted.

let L, be the Euclidean distance from the least squares point, p, to the true parameter point β . Then if λ , i = 1 . . . t are the eigenvalues of X'X.

$$E[L_1^2] = \sigma^2\Sigma(1/\lambda_i)$$

If one or more of the $\lambda_i are very small it is clear that \hat{\beta}, although unbiased, may be far removed from \beta$˙Choosing k *Ridge Trace* : One of the main obstacles in using ridge regression is in choosing an appropriate value of k.A plot is drawn to show the ridge regression coefficients as a function of k. When viewing the ridge trace, the analyst picks a value for k for which the regression coefficients have stabilized. Often, the regression coefficients will vary widely for small values of k and then stabilize. Choose the smallest value of k possible (which introduces the smallest bias) after which the regression coefficients have seem to remain constant. Note that increasing k will eventually drive the regression coefficients to zero.

Analytic K: This is an iterative method for selecting k. o obtain the first value of k, we

use the least squares coefficients. This produces a value of k. Using this new k, a new set of coefficients is found, and so on.

b
Answer: The table is created as explained. The complete explanation of the loading of data into MySQL is given in Appendix 1.

c i)
Answer: The histograms are plotted through R. For numerical data hist is used. For categorical data I have used plot with the type="h" which is histogram. Since the categorical data(make,model,type and trim) have lengthy names , in some graphs there is missing of these names. The code for histogram is written in **VehicleHitogram.R** . The output of the code which contain all histograms are present in **Histograms.pdf**

c ii)
Answer: The code for correlation is written in **correlation.R** . The output of the code which contain all histograms are present in **Correlation.pdf**

On the top right corner of each graph you find the correlation between two attributes. There are basically four types of data variables and the graph also varies based on it.

1) Continuous data vs Continuous data: Example : price vs mileage. Both the attributes consists of continuous data. For this I used "plot" API to plot the graph. This gives the distribution of price over mileage. The correlation is found by the formula "cor" present in R. I am rounding the correlation value to 2.
example: cor(price,mileage)

2) Continuous data vs categorical data: Example : price vs makers. Since the categorical data doesn't numerically correlate with continuous data "box plot" is drawn to understand the correlation between them. Box plot is given by the quartile ranges. If the quartile ranges, especially the median overlaps between categories then the correlation is said to be nearing to 0. Otherwise the correlation is said to be high(either nearing -1 or +1) . In price vs makers , Cadillac doesn't overlap with any other makers. Rest all makers overlap on each other but there is enough variation in their ranges. So we can come to conclusion that price is somewhat correlated to Model. In the case of Price vs model, all the models fall between same range. in this case we can conclude that there is no correlation between these price and Model.
3) Categorical data vs Categorical data: It is very difficult to analyze the correlation between two categorical data. the best way to go about is to draw a "Mosaic plot" between these two to see the correlation. To analyze a mosaic plot lets take an example of makers vs models. In this graph the first row which belong to particular model has only one maker. similarly all model have unique maker. In such situation the correlation is said to be one. In case of makers and type , type sedan is common among all the makers so there is no correlation.

4) discrete data vs (continuous , categorical,discrete): cylinders, liters, sound,cruise and leather were discrete data.

In-case of discrete vs continuous I have used box plot to analyze it and "cor" function to calculate correlation. Example would be cylinders vs Price.
In-case of discrete vs continuous I have used mosaic plot and best example would be cylinders vs makers.
I-case of discrete vs discrete again mosaic plot was better to visualize. sound vs cruise is an example.

The table of correlation is present in **correlation.bmp**. This is done using "corrplot" package Note the correlation of categorical data is not calculated but just analyzed. The analyzed report is the next column.

	Make	Model	Trim	Type
Make	Yes	Yes	No	No
Model	Yes	Yes	No	No
Trim	No	No	Yes	No
Trim	No	No	No	Yes

c iii)
Answer: By analyzing the correlation I have filtered the features. This method is called correlation feature selection.It states that "Good feature subsets contain features highly correlated with the classification, yet uncorrelated to each other". For the correlations please refer the previous question. According to the table in the previous question Price is correlated with make, model, trim, type, cylinders, liters and cruise. For rest of the things there was no significant correlation it with price. So now we have reduced to seven parameters. But In these seven parameters make is correlated with model and cylinders is correlated with liters. So I had choose either make or model, cylinders or liters. I have used dummy coding with attributes make,model,trim and type. This was possible with the help of package "psych". On finding R-squared of (make,price) and (model,price) I found that (model,price) R-squared is high therefore I have chosen model above make. Similarly with the case of cylinders and liters. Finally, I have gone with model and cylinders.

To code is written in **ModelComparision.R** and the output of the code is in **modelsdata.txt**. After removing unwanted features, there was still categorical data to deal with. Then all the possible combination of attributes are chosen and model matrix is created. For each possible model, results are calculated. Similarly adjusted R squared and P-value is calculated. The result is printed in the tabular format as present in the output file. The Anova output of combination of all models is also calculated and printed.

Analysis:

```
From the output of Anova we get P-value between models which tells that how to
compare two models.The p-value is less than 0.05
the models are very much similar to each other. In this comparison there
very less difference of RSS(residual sum of squares) between the models
and the p-value is very less(<0.05)

/*************************************************************/
Analysis of Variance Table

Model 1: price ~ 1 + model + trim + type + cylinders + cruise
Model 2: price ~ 1 + model + type + cylinders + cruise
  Res.Df        RSS  Df  Sum of Sq      F    Pr(>F)
1    733 2310530130
2    766 3209420957 -33 -898890827 8.6414 < 2.2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
/*************************************************************/

In this example there is huge difference in RSS(Residual Sum of Squares) and the p-value
is very high (> 0.05). There is no similarity between the models.

/*************************************************************/
```

```
Analysis of Variance Table

Model 1: price ~ 1 + model + trim + type + cylinders + cruise
Model 2: price ~ 1 + model + trim + cylinders
  Res.Df        RSS Df Sum of Sq      F Pr(>F)
1    733 2310530130
2    734 2311277697 -1   -747567 0.2372 0.6264
/*************************************************************/
```

c iii)
Answer: **p-Value**: In statistics p-Value helps in determining the significance of the results[3]. Hypothesis tests are used to test the validity of a claim that is made.This claim that's on trial, in essence, is called the null hypothesis. The alternative hypothesis is the one you would believe if the null hypothesis is concluded to be untrue.All hypothesis tests ultimately use a p-value to weigh the strength of the evidence (what the data are telling you about the population). The p-value is a number between 0 and 1 and interpreted in the following way:

A small p-value (typically ≤ 0.05) indicates strong evidence against the null hypothesis, so you reject the null hypothesis.

A large p-value (≥ 0.05) indicates weak evidence against the null hypothesis, so you fail to reject the null hypothesis.

p-values very close to the cutoff (0.05) are considered to be marginal (could go either way).

R^2 : R-squared is a statistical measure of how close the data are to the fitted regression line. It is also known as the coefficient of determination, or the coefficient of multiple determination for multiple regression. The definition of R-squared is fairly straight-forward; it is the percentage of the response variable variation that is explained by a linear model.[4]

R-squared = Explained variation / Total variation

R-squared is always between 0 and 100%:

0% indicates that the model explains none of the variability of the response data around its mean. 100% indicates that the model explains all the variability of the response data around its mean.

In general, the higher the R-squared, the better the model fits your data.

p-value: very small, R^2 *- very small*: When we are considering a huge data, there might be cases where R-squared is less but still p-Value is also less. In-fact in huge data there might be chances of this happening. Low R-squared doesn't mean the sample is bad. Moreover it depends on the domain of the data as well. In medical field the low R-squared is also given importance as it is very critical for them.
p-value: very large, R^2 *- very small*: The model which gives this data is serious anomalies and rejects null hypothesis. Moreover there is more variation of data around the mean. This model should be neglected.
p-value: very large, R^2 *- very large*: The model which gives this data is serious anomalies and rejects null hypothesis. Even though there is less variation from mean value p-value is more trusted compared to R^2.
p-value: very small, R^2 *- very large*: The model is the best model as there proves the null hypothesis

and the variation around the mean is also very less.

Problem 3

i

Answer: The extraction of data and putting into MySQL is explained in **Appendix 2**.

1) **Users**: (a) Zipcode in users is erroneous. Some Zipcodes are more than 99999. Since Zipcode is no where used in the agglomerative algorithm it is not affected. But still it is important to clean it as it might be used some time in later. Following SQL queries is used to detect zipcode errors and correct them.

```
mysql> select * from users1 where length(zipcode) > 5;
mysql> select zipcode from users1 where cast(zipcode as unsigned) > 99999;

This is the anomaly but we have values like  xxxxx-yyyy which is valid but not of great use. W
This is done because update table with condition > 99999 wont work if have data in the format

mysql> update users1 set zipcode = substring_index(zipcode,'-',1) where zipcode like "%-%";

Now make entries greater than 99999 to NULL(Don't know what to replace it with)

mysql> update users1 set zipcode = null where cast(zipcode as unsigned) > 99999;

(b) Checked errors with gender and found none:
mysql> select * from users1 where gender != "M" and gender != "F";
Empty set (1.83 sec)

(c) Checked errors for age but found none:
mysql> select * from users1 where age not in(1,18,25,35,45,50,56);
Empty set (0.01 sec)

(d) Checked errors for occupation:
mysql> select * from users1 where age = 1 and occupation in (3,5,6,7,11,13,14,16);
+--------+--------+------+------------+---------+
| userid | gender | age  | occupation | zipcode |
+--------+--------+------+------------+---------+
|   1468 | M      |    1 |         14 | 19147   |
|   4006 | M      |    1 |          7 | 48038   |
|   4413 | M      |    1 |         13 | 37130   |
|   4431 | M      |    1 |         14 | 80401   |
|   4787 | M      |    1 |         11 | 94610   |
|   5768 | F      |    1 |          7 | 20852   |
+--------+--------+------+------------+---------+
6 rows in set (0.03 sec)

Age with one cannot have any of these occupation. This will affect the agglomerative algorithm

mysql> update users1 set occupation = 10 where age = 1 and occupation in (3,5,6,7,11,13,14,16)
Query OK, 6 rows affected (0.62 sec)
Rows matched: 6  Changed: 6  Warnings: 0

mysql> select * from users1 where occupation = 10 and age not in(1,18);
```

```
Users with occupation 10 cannot be in any other age bracket apart from 1 and 18. If such thing

mysql> update users1 set age = 1 where occupation =10 and age not in (1,18);

2) \textbf{Movies}:
No Duplicates in movieid.

mysql> select movieid,count(*) as count from movies1 group by movieid having count(*) > 1;
Empty set (0.56 sec)

No Duplicates in title.

mysql> select title,count(*) as count from movies group by title having count(*)> 1;
Empty set (0.79 sec)

No empty year

mysql> select * from movies where title not like "%(%)%";
Empty set (0.01 sec)

No Genre names missing or misspelled

mysql> select * from movies1 where genre not like "%Action%"
and genre not like "%Adventure%" and genre not like "%Animation%"
and genre not like "%Children's%" and genre not like "%Comedy%"
and genre not like "%Crime%" and genre not like "%Documentary%" and genre not like "%Drama%" a
and genre not like "%Film-Noir%" and genre not like "%Horror%"
and genre not like "%Musical%" and genre not like "%Mystery%"
and genre not like "%Romance%" and genre not like "%Sci-Fi%"
and genre not like "%Thriller%" and genre not like "%War%"
and genre not like "%Western%";
Empty set (0.01 sec)
```

3) **Ratings**

```
Boundary checking for movieid , userid and rating done. No errors found. Boundaries of timesta

(a) Each user has rated atleast 20 movies. No errors there.

mysql>  select userid,count(*) as count from ratings1 group by userid having count(*) < 20;
Empty set (1.82 sec)

(b) Validation of userid from Users->userid. Validation succeeded

mysql>  select userid from ratings1 where userid not in (select userid from users);
Empty set (3.32 sec)

(c) Validation of movieid from Movies->movieid. Validation succeeded

mysql> select movieid from ratings1 where movieid not in (select movieid from movies1);
```

```
Empty set (2.42 sec)
```

```
I have found no more errors in the data.
```

ii
Answer: I have assumed computer scientists mean the people under occupation 15 which are scientists.

```
mysql> select count(case when M.rating > F.rating then 1 end) as "Male Rated High", count(case
+-----------------+-------------------+---------------------------+
| Male Rated High | Female Rated High | Male and Female rated same |
+-----------------+-------------------+---------------------------+
|           23026 |             25626 |                     27074 |
+-----------------+-------------------+---------------------------+
1 row in set (17.93 sec)
```

```
The above output says that for a given movie which is been rated by both male scientist and fe
```

iii
Answer:

```
mysql> select count(case when u.gender = "M" then 1 end) as "Males", count(case
when u.gender = "F" then 1 end) as "Females" from ratings r,users u where r.user
id = u.userid;
+--------+---------+
| Males  | Females |
+--------+---------+
| 753769 |  246440 |
+--------+---------+
1 row in set (8 min 33.55 sec)
```

```
This shows that men have rated more movies.
```

iv
Answer:

```
mysql> select count(case when m.genre like '%Action%' then 1 end) as "Action",
count(case when m.genre like '%Adventure%' then 1 end) as "Adventures",
count(case when m.genre like '%Animation%'then 1 end) as
"Animation",count(case when m.genre like '%Children%' then 1 end) as "Children",
count(case when m.genre like '%Comedy%' then 1 end) as "comedy", count(case
when m.genre like '%Crime%' then 1
end) as "Crime", count(case when m.genre like '%Documentary%' then 1 end) as "Documentary",
count(case when m.genre like '%Drama%'
then 1 end) as "Drama", count
(case when m.genre like '%Fantasy%' then 1 end) as "Fantasy",
count(case when m.genre like '%Film-Noir%' then 1 end) as
```

```
"Film-Noir", count(case when m.genre
like '%Horror%' then 1 end) as "Horror",
count(case when m.genre
like '%musical%' then 1 end) as "Musical", count(case when m.genre
like '%Mystery%' then 1 end) as"Mystery",
count(case when m.genre
like '%Romance%' then 1 end) as "Romance",
count(case when m.genre
like '%Sci-Fi%' then 1 end) as "Sci-Fi",
count(case when m.genre like '%Thriller%' then 1 end) as "Thriller", count(case when
m.genre like '%War%' then 1 end) as "War",
count(case when m.genre
like '%Western%' then 1 end) as
"Western" from ratings r,movies m
where r.movieid = m.movieid;
```

— Action — Adventures — Animation — Children — comedy — Crime — Documentary — Drama — Fantasy — Film-Noir — Horror — Musical — Mystery — Romance — Sci-Fi — Thriller — War — Western — ——— — 257457 — 133953 — 43293 — 72186 — 356580 — 79541 — 7910 — 354529 — 36301 — 18261 — 76386 — 41533 — 40178 — 147523 — 157294 — 189680 — 68527 — 20683 —

1 row in set (8 min 17.29 sec)

This shows that Comedy just edges out Drama movies.

iv

Answer: 1) The clustering on age and profession.

Before doing clustering lets analyse data with few graphs for few dataset.

Initial Analysis on **userinitialreading.R** and its output is written in **UserAnalysis.pdf**.

Create indexes to make joins faster as our queries contain joins

alter table ratings add index usermovie(userid,movieid);
alter table users add index userid(userid);
alter table movies add index movieid(movieid);

I have used java program to connect to MySQL and to do agglomerative Clustering. The java code is present in two files which is under the folder **Agglomerative1**

a) MySQLConnection.java: This basically for MySQL connection and execution of query and filling the data structure initially.

b) ReadData.java: Contains code for agglomeration and code to find similarity between genre and rating.

.

Algorithm steps

i) Connect to MySQL extract Query from it. Store all the tuples in collection class. Each object contain age,occupation, dictionary of genres, rating, two child links which are empty initially.

ii) In every iteration, find the two tulpes which is contributing to lowest distance calculated by distance formula.

iii)A parent node is created and average of age , mode of occupation, added dictionary of Genres and average rating is stored. The two nodes are linked as child nodes to this parent node.

iv)This parent node is added to the collection and the child nodes are deleted. this is done till there is only one node remaining.

For clustering I have distance formula which will be explained further. The technique used is Group Average: It means that proximity among all pairs of points in the different clusters. This is an intermediate approach between the single and complete link list.[5][6]

The code output is pasted in **Appendix 3** followed by its explanation of similarity.

The short note on running the program on BigRed2 (Super Computer) is given in **Appendix 5**

2) Clustering on genre and rating: The clustering code is similar to the above one with changes in query, distance formula and similarity calculations. The java code is present in two files which is under the folder **Agglomerative2**
a) MySQLConnection.java: This basically for MySQL connection and execution of query and filling the data structure initially.
b) ReadData.java: Contains code for agglomeration and code to find similarity between genre and rating.
.
Algorithm steps
i) Connect to MySQL extract Query from it. Store all the tuples in collection class. Each object contain age,occupation, dictionary of genres,gender, rating, two child links which are empty initially.
ii) In every iteration, find the two tulpes which is contributing to lowest distance calculated by distance formula.
iii)A parent node is created and average of age , mode of occupation, added dictionary of Genres and average rating is stored. The two nodes are linked as child nodes to this parent node.
iv)This parent node is added to the collection and the child nodes are deleted. this is done till there is only one node remaining.

The code output is pasted in **Appendix 4** followed by its explanation of similarity.

3) Distance function used for cluster on age and profession :

Cosine Similarity: Given two tuple t1 and t2 the cosine similarity based on age and profession is given by:

$$Distance = \frac{(t1.Age * t2.Age) + (t1.occupation * t2.occupation)}{\sqrt{t1.Age^2 + t1.occupation^2} * \sqrt{t2.Age^2 + t2.occupation^2}}$$

The advantage of this function is that there the cosine distance will give how the similarity between two tuples based on their variations. The higher the cosine value the better similar the two tuples are. But for finding out the common attribute for the parent I have used weighted mean for the age. Mode of the occupation that is followed. Since I am holding the total number of ratings tuples within a parent it is easier to calculate the weighed mean and mode.

Distance function used for cluster on genre and rating
I have used combination of two distance function.
Distance = Genre Jaccard Coefficient + Rating Ratio
$GenreJaccradCoefficient = \frac{commongenrespresentbetweentwonodes}{Totalnumbergenrespresentbyunionoftwonodes}$
$JaccardCoeffcient = \frac{A \cap B}{A \cup B}$
Rating Ratio between two nodes = 1 - (abs(R1.rating - R2.rating) / 4)
The sum of two distances never crosses value 2. lower the distance the similar are the nodes.

4) Quality of Clustering

Lot of SQL queries are performed to validate the results that are obtained. Few are quoted below as example.

(i)The Average rating obtained through clustering is in-fact correct.

```
mysql> SELECT avg(lim.rating) FROM (select u.age,u.occupation,r.rating from user
s1 u, movies1 m, ratings1 r where u.userid = r.userid and m.movieid = r.movieid
limit 1000) as lim;
+-----------------+
| avg(lim.rating) |
+-----------------+
|          3.7880 |
+-----------------+
1 row in set (0.01 sec)
```

APPENDIX

1. *Loading data from vehicles2.txt to MySQL.*
The data was separated by the delimiter "tab" or " ". It was very difficult to analyze the components of make,model,trim and type as all have spaces between them. There was a weird error in loading the data into MySQL which said that there is more parameters than given in the table veh_pri. To confirm it I replaced all " " by "," making it a CSV. It helped me in distinguishing between make,model,trim and type. Also I found that there was 6 commas(",") at the end of each tuple. I deleted all tail commas. This was done completely through notepad++ and the file name is attached as vehicles2.txt. Table is created as given in the question. The data is loaded into the Database through SQL command:

```
mysql>Load data infile 'vehicle2.txt' into table veh_pri fields terminated by ','; \newline
```

```
mysql> describe veh_pri;
+-----------+----------------------+------+-----+---------+-------+
| Field     | Type                 | Null | Key | Default | Extra |
+-----------+----------------------+------+-----+---------+-------+
| price     | decimal(8,2)         | YES  |     | NULL    |       |
| mileage   | mediumint(8) unsigned | YES  |     | NULL    |       |
| make      | varchar(20)          | YES  |     | NULL    |       |
| model     | varchar(20)          | YES  |     | NULL    |       |
| trim      | varchar(20)          | YES  |     | NULL    |       |
| type      | varchar(20)          | YES  |     | NULL    |       |
| cylinders | tinyint(3) unsigned  | YES  |     | NULL    |       |
| liters    | decimal(3,1)         | YES  |     | NULL    |       |
| doors     | tinyint(3) unsigned  | YES  |     | NULL    |       |
| cruise    | tinyint(1) unsigned  | YES  |     | NULL    |       |
| sound     | tinyint(1) unsigned  | YES  |     | NULL    |       |
| leather   | tinyint(1) unsigned  | YES  |     | NULL    |       |
+-----------+----------------------+------+-----+---------+-------+
12 rows in set (0.07 sec)
```

```
mysql> select count(*) from veh_pri;
+----------+
| count(*) |
+----------+
|      804 |
+----------+
```

```
1 row in set (0.36 sec)
```

```
mysql> select cylinders, count(cylinders) from veh_pri group by cylinders;
+-----------+------------------+
| cylinders | count(cylinders) |
+-----------+------------------+
|         4 |              394 |
|         6 |              310 |
|         8 |              100 |
+-----------+------------------+
3 rows in set (0.14 sec)
```

Note: I have replaced the type of cruise,sound and leather from bit(1) to tinyint(1) unsigned as there was some problem with my MySQL version compatibility with bit type. The documentation said they are of same type and either can be used.

2.*Loading data from movies.dat,ratings.dat and users.dat to MySQL* : The following queries are performed to load the given data into MySQL.

```
mysql> create table users1(userid smallint(2) unsigned,gender varchar(1),age tinyint(1) unsigned, occupa

mysql> load data infile 'users.dat' into table users1 fields terminated by '::';

mysql> create table movies1(movieid smallint(2) unsigned,title varchar(100),genre varchar(50));

mysql> load data infile 'movies.dat' into table movies1 fields terminated by '::';

mysql> create table ratings1(userid smallint(2) unsigned,movieid smallint(2) unsigned, rating tinyint(1

mysql> load data infile 'ratings.dat' into table ratings1 fields terminated by '::';

mysql> describe users1;
+------------+----------------------+------+-----+---------+-------+
| Field      | Type                 | Null | Key | Default | Extra |
+------------+----------------------+------+-----+---------+-------+
| userid     | smallint(2) unsigned | YES  | MUL | NULL    |       |
| gender     | varchar(1)           | YES  |     | NULL    |       |
| age        | tinyint(1)           | YES  |     | NULL    |       |
| occupation | smallint(2) unsigned | YES  |     | NULL    |       |
| zipcode    | varchar(20)          | YES  |     | NULL    |       |
+------------+----------------------+------+-----+---------+-------+
5 rows in set (0.20 sec)

mysql> describe ratings1;
+-----------+----------------------+------+-----+---------+-------+
| Field     | Type                 | Null | Key | Default | Extra |
+-----------+----------------------+------+-----+---------+-------+
| userid    | smallint(2) unsigned | YES  | MUL | NULL    |       |
| movieid   | smallint(2) unsigned | YES  |     | NULL    |       |
| rating    | tinyint(1) unsigned  | YES  |     | NULL    |       |
| timestamp | int(4)               | YES  |     | NULL    |       |
```

```
+-----------+----------------------+------+-----+---------+-------+
4 rows in set (0.06 sec)

mysql> describe movies1;
+---------+----------------------+------+-----+---------+-------+
| Field   | Type                 | Null | Key | Default | Extra |
+---------+----------------------+------+-----+---------+-------+
| movieid | smallint(2) unsigned | YES  | MUL | NULL    |       |
| title   | varchar(100)         | YES  |     | NULL    |       |
| genre   | varchar(50)          | YES  |     | NULL    |       |
+---------+----------------------+------+-----+---------+-------+
3 rows in set (0.10 sec)
```

The joins were taking longer time because of the huge data. I have partitioned the data to make parallel executions in MySQL. Users are partitioned on the basis of userid. Movies are partitioned based on movieid. Ratings are partitioned on both userid and movieid.

```
mysql> alter table users1 partition by range(userid)
    -> (partition p0 values less than (500),
    -> partition p1 values less than (1000),
    -> partition p2 values less than (1500),
    -> partition p3 values less than (2000),
    -> partition p4 values less than (2500),
    -> partition p5 values less than (3000),
    -> partition p6 values less than (3500),
    -> partition p7 values less than (4000),
    -> partition p8 values less than (4500),
    -> partition p9 values less than (5000),
    -> partition p10 values less than (5500),
    -> partition p11 values less than MAXVALUE);


mysql> alter table ratings1 partition by range(userid)
    -> (partition p0 values less than (500),
    -> partition p1 values less than (1000),
    -> partition p2 values less than (1500),
    -> partition p3 values less than (2000),
    -> partition p4 values less than (2500),
    -> partition p5 values less than (3000),
    -> partition p6 values less than (3500),
    -> partition p7 values less than (4000),
    -> partition p8 values less than (4500),
    -> partition p9 values less than (5000),
    -> partition p10 values less than (5500),
    -> partition p11 values less than MAXVALUE);

mysql> alter table movies1 partition by range(movieid)
    -> (partition p0 values less than (500),
    -> partition p1 values less than (1000),
    -> partition p2 values less than (1500),
    -> partition p3 values less than (2000),
    -> partition p4 values less than (2500),
    -> partition p5 values less than (3000),
```

```
    -> partition p6 values less than (3500),
    -> partition p7 values less than MAXVALUE);

mysql> alter table ratings1 partition by range(movieid)
    -> (partition p0 values less than (500),
    -> partition p1 values less than (1000),
    -> partition p2 values less than (1500),
    -> partition p3 values less than (2000),
    -> partition p4 values less than (2500),
    -> partition p5 values less than (3000),
    -> partition p6 values less than (3500),
    -> partition p7 values less than MAXVALUE);
```

3. *Agglomerative clustering on Age,profession*

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.ArrayList;
import java.util.HashMap;

public class MySQLConnection
{
private Statement _stmt;
private Connection _conn;

public MySQLConnection(String dataBase,String username,String password)
{
String JDBCDRIVER = "com.mysql.jdbc.Driver";
String DB_URL = "jdbc:mysql://localhost/"+dataBase;

_conn = null;

try{

      Class.forName("com.mysql.jdbc.Driver");
      //System.out.println("Connecting to database...");
      _conn = DriverManager.getConnection(DB_URL,username,password);

      //System.out.println("Creating statement...");
      _stmt = _conn.createStatement();
}
catch(SQLException se)
   {

      se.printStackTrace();
   }catch(Exception e){

      e.printStackTrace();
   }
}
```

```
public boolean getData(String sql,ArrayList<tuple> tuples)
{
boolean status  = false;
ResultSet rs=null;

try {
rs = _stmt.executeQuery(sql);

status = (rs == null)?false:true;

int limitedTo = 1000;
while(rs.next() && limitedTo-- > 0)
{
tuple t = new tuple();
t.age = rs.getInt(1);
t.occupation = rs.getInt(2);

t.genre = new HashMap<String,Integer>();
String gen= rs.getString(3);

for(String S: gen.split("\\|"))
{
  if(!t.genre.containsKey(S))
t.genre.put(S,1);
else
t.genre.put(S, t.genre.get(S) + 1);
}

t.rating = rs.getDouble(4);
t.totalTuples = 1;
tuples.add(t);
}
rs.close();

} catch (SQLException e) {
// TODO Auto-generated catch block
e.printStackTrace();
}

return status;
}

void stop()
{
      try {
_stmt.close();
_conn.close();
      } catch (SQLException e) {

e.printStackTrace();
}
}
```

```
};

import java.sql.*;
import java.util.ArrayList;
import java.util.Collections;
import java.util.HashMap;
import java.util.Iterator;
import java.util.LinkedList;
import java.util.Map;
import java.util.Queue;
import java.util.Set;


class tuple
{
public int age;
public int occupation;
public HashMap<String, Integer> genre;
public double rating;
tuple llink;
tuple rlink;
int totalTuples;
};




public class ReadData
{
public static double normalizedAgeHop(int age1,int age2)
{
if(age1 < age2)
{
int t = age1;
age1 = age2;
age2 = t;
}

int hop = 0;
while(age1 < age2)
{
hop++;
switch(age1)
{
case 1: age1 = 18;
break;
case 18: age1 = 25;
break;
case 25: age1 = 35;
break;
```

```
case 35: age1 = 45;
break;
case 45: age1 = 50;
break;
case 56: age1 = 56;
break;
}
}
return hop/6;
}

public static void agglomerative(ArrayList<tuple> tuples)
{
int size = tuples.size()-1;
for(int i = 0; i < size; ++i)
{
//Getting the tuples with minimum distance and getting the distance in terms of Age and occupation

int numberOfTuples = tuples.size();
double maxCosine = 0.0;
tuple left = null,right = null;

for(int j = 0; j< numberOfTuples;++j)
{
tuple t1 = tuples.get(j);
for(int k = j+1; k< numberOfTuples;++k)
{
tuple t2 = tuples.get(k);
//double distance = (normalizedAgeHop(t1.age,t2.age) + ((t1.occupation == t2.occupation)?0:1))/2;
double cosine = ((t1.age * t2.age) + (t1.occupation * t2.occupation))/
(Math.sqrt(t1.age*t1.age + t1.occupation * t1.occupation)*Math.sqrt(t2.age*t2.age + t2.occupation * t2.
if(cosine > maxCosine)
{
maxCosine = cosine;
left = t1;
right = t2;
}
}
}


tuple clusterParent = new tuple();
clusterParent.totalTuples = left.totalTuples + right.totalTuples;
clusterParent.age = (left.age * left.totalTuples + right.totalTuples*right.age)/(left.totalTuples + rig

clusterParent.occupation = left.totalTuples>right.totalTuples?left.occupation : right.occupation;

clusterParent.llink = new tuple();
clusterParent.rlink = new tuple();
clusterParent.llink = left;
clusterParent.rlink = right;

String[] S = left.genre.keySet().toArray(new String[left.genre.size()]);
```

```
clusterParent.genre = new HashMap<String,Integer>();
for(int loc =0 ;loc< S.length; ++loc)
{
if(!clusterParent.genre.containsKey(S[loc]))
clusterParent.genre.put(S[loc],left.genre.get(S[loc]));
else
clusterParent.genre.put(S[loc], clusterParent.genre.get(S[loc]) + left.genre.get(S[loc]));
}

String[] S1 = right.genre.keySet().toArray(new String[right.genre.size()]);
for(int loc =0 ;loc< S1.length; ++loc)
{
if(!clusterParent.genre.containsKey(S1[loc]))
clusterParent.genre.put(S1[loc],right.genre.get(S1[loc]));
else
clusterParent.genre.put(S1[loc], clusterParent.genre.get(S1[loc]) + right.genre.get(S1[loc]));
}

clusterParent.rating = (left.rating * left.totalTuples + right.rating * right.totalTuples)/(left.totalTu

tuples.remove(left);
tuples.remove(right);

tuples.add(clusterParent);
// System.out.println("Iteration = "+i+"  size of tuples = "+tuples.size());
}

System.out.println("Done");
}

public static void SimilarityGenresandRating(ArrayList<tuple> tuples)
{
//Travelling 3 levels for checking the similarity

Queue q = new LinkedList<tuple>();
q.add(tuples.get(0));

int level = 1;
while(!q.isEmpty())
{
tuple t= (tuple) q.remove();
System.out.println("*****************At Level = "+Math.ceil(Math.log(level))+" Age  ="+t.age+" Occupatio
System.out.println("Genres of movie they watch will be in the ratio of =");

int totalCount = 0;

Iterator it = t.genre.entrySet().iterator();
while(it.hasNext())
{
Map.Entry pairs = (Map.Entry)it.next();

totalCount += (int) pairs.getValue();
}
```

```
it = t.genre.entrySet().iterator();
while(it.hasNext())
{
Map.Entry pairs = (Map.Entry)it.next();

System.out.println("Ratio of watching   "+pairs.getKey()+" movie is = "+(Double.parseDouble(pairs.getVa
}

System.out.println("Average rating of the users in this age group id = "+t.rating);
if(level<8)
{
++level;
if(t.llink != null)
q.add(t.llink);
if(t.rlink != null)
q.add(t.rlink);
}
}
}

public static void main(String[] args)
{
String sql;
    sql = "SELECT u.age,u.occupation,m.genre,r.rating FROM users1 u, movies1 m, ratings1 r where " +
       "u.userid = r.userid and m.movieid = r.movieid";
    ArrayList<tuple> tuples = new ArrayList<tuple>();

    MySQLConnection mysqlConnection = new MySQLConnection("suhas_db", "root", "root");
    boolean status = mysqlConnection.getData(sql, tuples);

if(status == false)
{
System.out.println("Wrong in data fetch");
System.exit(0);
}
else
{
System.out.println("Total Tuple Strength = "+tuples.size());
}

agglomerative(tuples);

mysqlConnection.stop();

SimilarityGenresandRating(tuples);

}

}
```

The algorithm takes up lot of memory thereby no completing for all the one million plus records of rating.
I have put here the sample output for first 1000 ratings.

I have printed nodes of top three levels of the agglomerative cluster. At the root node it says that average age of people is 31. Their common occupation would be 16(Self-employed) and their ration of watching different genres of movie is given. Along with this their average rating is also given(3.788). The similarity that we can express here that people with age group of 31 are usually self employed and watch high percentage of Drama movies and their average rating would be 3.788.
Similarly at level 1 we can express similarity as people with average age group 1(¡18) have common occupation of 10(K-12 student) and watch high percentage of children movies with an average rating of 4.188.
A cross reference to verify the results is done through SQL and the output matches.

```
/*****************************************************************/
*****************At Level = 0.0 Age  =31 Occupation = 16
Genres of movie they watch will be in the ratio of =
Ratio of watching   Film-Noir movie is = 0.004215456674473068
Ratio of watching   Western movie is = 0.012646370023419205
Ratio of watching   Crime movie is = 0.03653395784543326
Ratio of watching   Action movie is = 0.13021077283372365
Ratio of watching   Adventure movie is = 0.06416861826697892
Ratio of watching   Romance movie is = 0.08946135831381732
Ratio of watching   Animation movie is = 0.03044496487119438
Ratio of watching   Fantasy movie is = 0.010304449648711944
Ratio of watching   War movie is = 0.03372365339578454
Ratio of watching   Horror movie is = 0.013583138173302109
Ratio of watching   Children's movie is = 0.03934426229508197
Ratio of watching   Musical movie is = 0.02857142857142857
Ratio of watching   Sci-Fi movie is = 0.05667447306791569
Ratio of watching   Comedy movie is = 0.13302107728337237
Ratio of watching   Thriller movie is = 0.08196721311475409
Ratio of watching   Mystery movie is = 0.0117096018735363
Ratio of watching   Drama movie is = 0.21920374707259954
Ratio of watching   Documentary movie is = 0.004215456674473068
Average rating of the users in this age group id = 3.788
*****************At Level = 1.0 Age  =1 Occupation = 10
Genres of movie they watch will be in the ratio of =
Ratio of watching   Crime movie is = 0.017241379310344827
Ratio of watching   Adventure movie is = 0.04310344827586207
Ratio of watching   Action movie is = 0.04310344827586207
Ratio of watching   War movie is = 0.017241379310344827
Ratio of watching   Fantasy movie is = 0.02586206896551724
Ratio of watching   Animation movie is = 0.15517241379310345
Ratio of watching   Romance movie is = 0.05172413793103448
Ratio of watching   Children's movie is = 0.1724137931034483
Ratio of watching   Sci-Fi movie is = 0.02586206896551724
Ratio of watching   Musical movie is = 0.1206896551724138
Ratio of watching   Comedy movie is = 0.1206896551724138
Ratio of watching   Thriller movie is = 0.02586206896551724
Ratio of watching   Drama movie is = 0.1810344827586207
Average rating of the users in this age group id = 4.188679245283019
*****************At Level = 2.0 Age  =33 Occupation = 16
Genres of movie they watch will be in the ratio of =
Ratio of watching   Film-Noir movie is = 0.004457652303120356
Ratio of watching   Crime movie is = 0.037642397226349676
Ratio of watching   Animation movie is = 0.023278850916295196
Ratio of watching   Romance movie is = 0.09162951956414066
```

```
Ratio of watching   Horror movie is = 0.014363546310054482
Ratio of watching   Comedy movie is = 0.1337295690936107
Ratio of watching   Mystery movie is = 0.012382367508667657
Ratio of watching   Western movie is = 0.01337295690936107
Ratio of watching   Adventure movie is = 0.06537890044576523
Ratio of watching   Action movie is = 0.1352154531946508
Ratio of watching   Fantasy movie is = 0.009410599306587419
Ratio of watching   War movie is = 0.03467062902426944
Ratio of watching   Children's movie is = 0.0316988608221892
Ratio of watching   Sci-Fi movie is = 0.05844477464091134
Ratio of watching   Musical movie is = 0.023278850916295196
Ratio of watching   Thriller movie is = 0.08519068845963348
Ratio of watching   Drama movie is = 0.2213967310549777
Ratio of watching   Documentary movie is = 0.004457652303120356
Average rating of the users in this age group id = 3.765575501583949
*****************At Level = 2.0 Age  =1 Occupation = 10
Genres of movie they watch will be in the ratio of =
Ratio of watching   Crime movie is = 0.0425531914893617
Ratio of watching   Romance movie is = 0.0425531914893617
Ratio of watching   Animation movie is = 0.14893617021276595
Ratio of watching   Comedy movie is = 0.14893617021276595
Ratio of watching   Action movie is = 0.0425531914893617
Ratio of watching   Adventure movie is = 0.02127659574468085
Ratio of watching   Fantasy movie is = 0.0425531914893617
Ratio of watching   War movie is = 0.02127659574468085
Ratio of watching   Children's movie is = 0.1702127659574468
Ratio of watching   Musical movie is = 0.0851063829787234
Ratio of watching   Sci-Fi movie is = 0.02127659574468085
Ratio of watching   Thriller movie is = 0.0425531914893617
Ratio of watching   Drama movie is = 0.1702127659574468
Average rating of the users in this age group id = 4.285714285714286
*****************At Level = 2.0 Age  =1 Occupation = 10
Genres of movie they watch will be in the ratio of =
Ratio of watching   Adventure movie is = 0.057971014492753624
Ratio of watching   Action movie is = 0.043478260869565216
Ratio of watching   War movie is = 0.014492753623188406
Ratio of watching   Romance movie is = 0.057971014492753624
Ratio of watching   Animation movie is = 0.15942028985507245
Ratio of watching   Fantasy movie is = 0.014492753623188406
Ratio of watching   Children's movie is = 0.17391304347826086
Ratio of watching   Musical movie is = 0.14492753623188406
Ratio of watching   Sci-Fi movie is = 0.028985507246376812
Ratio of watching   Comedy movie is = 0.10144927536231885
Ratio of watching   Thriller movie is = 0.014492753623188406
Ratio of watching   Drama movie is = 0.18840579710144928
Average rating of the users in this age group id = 4.125
*****************At Level = 2.0 Age  =35 Occupation = 1
Genres of movie they watch will be in the ratio of =
Ratio of watching   Film-Noir movie is = 0.0074211502782931356
Ratio of watching   Crime movie is = 0.02040816326530612
Ratio of watching   Animation movie is = 0.03339517625231911
Ratio of watching   Romance movie is = 0.07792207792207792
Ratio of watching   Horror movie is = 0.016697588126159554
```

```
Ratio of watching   Comedy movie is = 0.15213358070500926
Ratio of watching   Mystery movie is = 0.014842300556586271
Ratio of watching   Western movie is = 0.00927643784786642
Ratio of watching   Adventure movie is = 0.09090909090909091
Ratio of watching   Action movie is = 0.14285714285714285
Ratio of watching   Fantasy movie is = 0.02040816326530612
Ratio of watching   War movie is = 0.04081632653061224
Ratio of watching   Children's movie is = 0.055658627087198514
Ratio of watching   Musical movie is = 0.04267161410018553
Ratio of watching   Sci-Fi movie is = 0.08719851576994433
Ratio of watching   Thriller movie is = 0.06307977736549165
Ratio of watching   Drama movie is = 0.12244897959183673
Ratio of watching   Documentary movie is = 0.0018552875695732839
Average rating of the users in this age group id = 4.155172413793103
******************At Level = 2.0 Age  =33 Occupation = 16
Genres of movie they watch will be in the ratio of =
Ratio of watching   Film-Noir movie is = 0.0033783783783783786
Ratio of watching   Crime movie is = 0.04391891891891892
Ratio of watching   Romance movie is = 0.09662162162162162
Ratio of watching   Animation movie is = 0.019594594594594596
Ratio of watching   Horror movie is = 0.013513513513513514
Ratio of watching   Comedy movie is = 0.12702702702702703
Ratio of watching   Mystery movie is = 0.011486486486486487
Ratio of watching   Western movie is = 0.014864864864864866
Ratio of watching   Adventure movie is = 0.056081081081081084
Ratio of watching   Action movie is = 0.13243243243243244
Ratio of watching   Fantasy movie is = 0.005405405405405406
Ratio of watching   War movie is = 0.032432432432432434
Ratio of watching   Children's movie is = 0.022972972972972974
Ratio of watching   Musical movie is = 0.016216216216216217
Ratio of watching   Sci-Fi movie is = 0.047972972972972976
Ratio of watching   Thriller movie is = 0.09324324324324325
Ratio of watching   Drama movie is = 0.2574324324324324
Ratio of watching   Documentary movie is = 0.005405405405405406
Average rating of the users in this age group id = 3.639160839160839
******************At Level = 3.0 Age  =1 Occupation = 10
Genres of movie they watch will be in the ratio of =
Ratio of watching   Crime movie is = 0.07692307692307693
Ratio of watching   Action movie is = 0.07692307692307693
Ratio of watching   Fantasy movie is = 0.07692307692307693
Ratio of watching   Romance movie is = 0.07692307692307693
Ratio of watching   Animation movie is = 0.07692307692307693
Ratio of watching   Children's movie is = 0.07692307692307693
Ratio of watching   Comedy movie is = 0.3076923076923077
Ratio of watching   Thriller movie is = 0.07692307692307693
Ratio of watching   Drama movie is = 0.15384615384615385
Average rating of the users in this age group id = 4.125
******************At Level = 3.0 Age  =1 Occupation = 10
Genres of movie they watch will be in the ratio of =
Ratio of watching   Crime movie is = 0.029411764705882353
Ratio of watching   Adventure movie is = 0.029411764705882353
Ratio of watching   Action movie is = 0.029411764705882353
Ratio of watching   War movie is = 0.029411764705882353
```

```
Ratio of watching   Fantasy movie is = 0.029411764705882353
Ratio of watching   Animation movie is = 0.17647058823529413
Ratio of watching   Romance movie is = 0.029411764705882353
Ratio of watching   Children's movie is = 0.20588235294117646
Ratio of watching   Sci-Fi movie is = 0.029411764705882353
Ratio of watching   Musical movie is = 0.11764705882352941
Ratio of watching   Comedy movie is = 0.08823529411764706
Ratio of watching   Thriller movie is = 0.029411764705882353
Ratio of watching   Drama movie is = 0.17647058823529413
Average rating of the users in this age group id = 4.384615384615385
*****************At Level = 3.0 Age  =1 Occupation = 10
Genres of movie they watch will be in the ratio of =
Ratio of watching   Action movie is = 0.02631578947368421
Ratio of watching   Adventure movie is = 0.05263157894736842
Ratio of watching   Fantasy movie is = 0.02631578947368421
Ratio of watching   Animation movie is = 0.15789473684210525
Ratio of watching   Romance movie is = 0.05263157894736842
Ratio of watching   Children's movie is = 0.18421052631578946
Ratio of watching   Sci-Fi movie is = 0.02631578947368421
Ratio of watching   Musical movie is = 0.23684210526315788
Ratio of watching   Comedy movie is = 0.07894736842105263
Ratio of watching   Thriller movie is = 0.02631578947368421
Ratio of watching   Drama movie is = 0.13157894736842105
Average rating of the users in this age group id = 4.0
*****************At Level = 3.0 Age  =1 Occupation = 10
Genres of movie they watch will be in the ratio of =
Ratio of watching   Adventure movie is = 0.06451612903225806
Ratio of watching   Action movie is = 0.06451612903225806
Ratio of watching   War movie is = 0.03225806451612903
Ratio of watching   Animation movie is = 0.16129032258064516
Ratio of watching   Romance movie is = 0.06451612903225806
Ratio of watching   Children's movie is = 0.16129032258064516
Ratio of watching   Musical movie is = 0.03225806451612903
Ratio of watching   Sci-Fi movie is = 0.03225806451612903
Ratio of watching   Comedy movie is = 0.12903225806451613
Ratio of watching   Drama movie is = 0.25806451612903225
Average rating of the users in this age group id = 4.25
*****************At Level = 3.0 Age  =35 Occupation = 1
Genres of movie they watch will be in the ratio of =
Ratio of watching   Film-Noir movie is = 0.008064516129032258
Ratio of watching   Crime movie is = 0.024193548387096774
Ratio of watching   Animation movie is = 0.016129032258064516
Ratio of watching   Romance movie is = 0.06854838709677419
Ratio of watching   Horror movie is = 0.012096774193548387
Ratio of watching   Comedy movie is = 0.16129032258064516
Ratio of watching   Mystery movie is = 0.008064516129032258
Ratio of watching   Western movie is = 0.004032258064516129
Ratio of watching   Adventure movie is = 0.0846774193548387
Ratio of watching   Action movie is = 0.18951612903225806
Ratio of watching   Fantasy movie is = 0.024193548387096774
Ratio of watching   War movie is = 0.036290322580645164
Ratio of watching   Children's movie is = 0.04032258064516129
Ratio of watching   Sci-Fi movie is = 0.10080645161290322
```

```
Ratio of watching   Musical movie is = 0.020161290322580645
Ratio of watching   Thriller movie is = 0.08870967741935484
Ratio of watching   Drama movie is = 0.10887096774193548
Ratio of watching   Documentary movie is = 0.004032258064516129
Average rating of the users in this age group id = 4.115384615384615
*****************At Level = 3.0 Age  =35 Occupation = 1
Genres of movie they watch will be in the ratio of =
Ratio of watching   Film-Noir movie is = 0.006872852233676976
Ratio of watching   Crime movie is = 0.01718213058419244
Ratio of watching   Romance movie is = 0.0859106529209622
Ratio of watching   Animation movie is = 0.048109965635738834
Ratio of watching   Horror movie is = 0.020618556701030927
Ratio of watching   Comedy movie is = 0.14432989690721648
Ratio of watching   Mystery movie is = 0.020618556701030927
Ratio of watching   Western movie is = 0.013745704467353952
Ratio of watching   Action movie is = 0.10309278350515463
Ratio of watching   Adventure movie is = 0.09621993127147767
Ratio of watching   Fantasy movie is = 0.01718213058419244
Ratio of watching   War movie is = 0.044673539518900345
Ratio of watching   Children's movie is = 0.06872852233676977
Ratio of watching   Musical movie is = 0.061855670103092786
Ratio of watching   Sci-Fi movie is = 0.07560137457044673
Ratio of watching   Thriller movie is = 0.041237113402061855
Ratio of watching   Drama movie is = 0.13402061855670103
Average rating of the users in this age group id = 4.1875
*****************At Level = 3.0 Age  =25 Occupation = 20
Genres of movie they watch will be in the ratio of =
Ratio of watching   Film-Noir movie is = 0.005873715124816446
Ratio of watching   Crime movie is = 0.05433186490455213
Ratio of watching   Romance movie is = 0.07782672540381791
Ratio of watching   Animation movie is = 0.023494860499265784
Ratio of watching   Horror movie is = 0.020558002936857563
Ratio of watching   Comedy movie is = 0.16593245227606462
Ratio of watching   Mystery movie is = 0.014684287812041116
Ratio of watching   Western movie is = 0.011747430249632892
Ratio of watching   Adventure movie is = 0.06020558002936858
Ratio of watching   Action movie is = 0.10866372980910426
Ratio of watching   Fantasy movie is = 0.004405286343612335
Ratio of watching   War movie is = 0.022026431718061675
Ratio of watching   Children's movie is = 0.020558002936857563
Ratio of watching   Sci-Fi movie is = 0.04405286343612335
Ratio of watching   Musical movie is = 0.005873715124816446
Ratio of watching   Thriller movie is = 0.10425844346549193
Ratio of watching   Drama movie is = 0.24375917767988253
Ratio of watching   Documentary movie is = 0.011747430249632892
Average rating of the users in this age group id = 3.4309859154929576
*****************At Level = 3.0 Age  =41 Occupation = 16
Genres of movie they watch will be in the ratio of =
Ratio of watching   Film-Noir movie is = 0.0012515644555694619
Ratio of watching   Crime movie is = 0.03504380475594493
Ratio of watching   Romance movie is = 0.11264080100125157
Ratio of watching   Animation movie is = 0.016270337922403004
Ratio of watching   Horror movie is = 0.007509386733416771
```

```
Ratio of watching   Comedy movie is = 0.09386733416770963
Ratio of watching   Mystery movie is = 0.008760951188986232
Ratio of watching   Western movie is = 0.017521902377972465
Ratio of watching   Action movie is = 0.15269086357947434
Ratio of watching   Adventure movie is = 0.052565707133917394
Ratio of watching   Fantasy movie is = 0.006257822277847309
Ratio of watching   War movie is = 0.04130162703379224
Ratio of watching   Children's movie is = 0.025031289111389236
Ratio of watching   Sci-Fi movie is = 0.05131414267834793
Ratio of watching   Musical movie is = 0.025031289111389236
Ratio of watching   Thriller movie is = 0.08385481852315395
Ratio of watching   Drama movie is = 0.2690863579474343
Average rating of the users in this age group id = 3.8444444444444446
/*****************************************************************/
```

4. *Agglomerative clustering on Genre,rating*

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.ArrayList;
import java.util.HashMap;

public class MySQLConnection
{
private Statement _stmt;
private Connection _conn;

public MySQLConnection(String dataBase,String username,String password)
{
String JDBCDRIVER = "com.mysql.jdbc.Driver";
String DB_URL = "jdbc:mysql://localhost/"+dataBase;

_conn = null;

try{

      Class.forName("com.mysql.jdbc.Driver");
      //System.out.println("Connecting to database...");
      _conn = DriverManager.getConnection(DB_URL,username,password);

      //System.out.println("Creating statement...");
      _stmt = _conn.createStatement();
}
catch(SQLException se)
   {

      se.printStackTrace();
   }catch(Exception e){

      e.printStackTrace();
   }
```

```
}

public boolean getData(String sql,ArrayList<tuple> tuples)
{
boolean status  = false;
ResultSet rs=null;

try {
rs = _stmt.executeQuery(sql);

status = (rs == null)?false:true;

int limitedTo = 1000;
while(rs.next() && limitedTo-- > 0)
{
tuple t = new tuple();
t.age = rs.getInt(3);

if(rs.getString(4).compareTo("M") == 0)
t.male=1;
else
t.female = 1;
t.occupation = rs.getInt(5);

t.genre = new HashMap<String,Integer>();
String gen= rs.getString(1);

for(String S: gen.split("\\|"))
{
  if(!t.genre.containsKey(S))
t.genre.put(S,1);
else
t.genre.put(S, t.genre.get(S) + 1);
}

t.rating = rs.getDouble(2);
t.totalTuples = 1;
tuples.add(t);
}
rs.close();

} catch (SQLException e) {
// TODO Auto-generated catch block
e.printStackTrace();
}

return status;
}

void stop()
{
      try {
_stmt.close();
```

```
_conn.close();
      } catch (SQLException e) {

e.printStackTrace();
}
}


};

import java.sql.*;
import java.util.ArrayList;
import java.util.Collections;
import java.util.HashMap;
import java.util.Iterator;
import java.util.LinkedList;
import java.util.Map;
import java.util.Map.Entry;
import java.util.Queue;
import java.util.Set;


class tuple
{
public int age;
public int occupation;
public HashMap<String, Integer> genre;
public double rating;
tuple llink;
tuple rlink;
int totalTuples;
int male;
int female;
};

public class ReadData
{
public static void agglomerative(ArrayList<tuple> tuples)
{
int size = tuples.size()-1;
for(int i = 0; i < size; ++i)
{
//Getting the tuples with minimum distance and getting the distance in terms of Age and occupation

int numberOfTuples = tuples.size();
double minDistance = Double.MAX_VALUE;
tuple left = null,right = null;

for(int j = 0; j< numberOfTuples;++j)
{
tuple t1 = tuples.get(j);

for(int k = j+1; k< numberOfTuples;++k)
```

```
{
tuple t2 = tuples.get(k);
//double distance = (normalizedAgeHop(t1.age,t2.age) + ((t1.occupation == t2.occupation)?0:1))/2;

HashMap<String, Integer> interGenre = new HashMap<String, Integer>();

String[] S;
if(t1.genre.keySet().toArray(new String[t1.genre.size()]).length < t2.genre.keySet().toArray(new String
S = t2.genre.keySet().toArray(new String[t2.genre.size()]);
else
S = t1.genre.keySet().toArray(new String[t1.genre.size()]);

int total = 0,common = 0;
for(int loc =0 ;loc< S.length; ++loc)
{
if(t1.genre.containsKey(S[loc]) && t2.genre.containsKey(S[loc]))
{
interGenre.put(S[loc],Math.min(t1.genre.get(S[loc]), t2.genre.get(S[loc])));
common+=Math.min(t1.genre.get(S[loc]), t2.genre.get(S[loc]));
total+=t1.genre.get(S[loc]) + t2.genre.get(S[loc]);
}
else if(!t1.genre.containsKey(S[loc]) && t2.genre.containsKey(S[loc]))
total+= t2.genre.get(S[loc]);
else if(t1.genre.containsKey(S[loc]) && !t2.genre.containsKey(S[loc]))
total+= t1.genre.get(S[loc]);

}

double distance = 1-Math.abs(t1.rating - t2.rating)/4;
if(total!=0)
distance+=(common* 1.0)/total;

if(minDistance > distance)
{
minDistance = distance;
left = t1;
right = t2;
}
}
}


tuple clusterParent = new tuple();
clusterParent.totalTuples = left.totalTuples + right.totalTuples;
clusterParent.age = (left.age * left.totalTuples + right.totalTuples*right.age)/(left.totalTuples + rig

clusterParent.occupation = left.totalTuples>right.totalTuples?left.occupation : right.occupation;

clusterParent.llink = new tuple();
clusterParent.rlink = new tuple();
clusterParent.llink = left;
clusterParent.rlink = right;
```

```
String[] S;
if(left.genre.keySet().toArray(new String[left.genre.size()]).length < right.genre.keySet().toArray(new
S = right.genre.keySet().toArray(new String[right.genre.size()]);
else
S = left.genre.keySet().toArray(new String[left.genre.size()]);

clusterParent.genre = new HashMap<String,Integer>();
for(int loc =0 ;loc< S.length; ++loc)
{
if(left.genre.containsKey(S[loc]) && right.genre.containsKey(S[loc]))
clusterParent.genre.put(S[loc],Math.min(left.genre.get(S[loc]), right.genre.get(S[loc])));
}

clusterParent.male = left.male + right.male;
clusterParent.female = left.female + right.female;

clusterParent.rating = (left.rating * left.totalTuples + right.rating * right.totalTuples)/(left.totalTu

tuples.remove(left);
tuples.remove(right);

tuples.add(clusterParent);
// System.out.println("Iteration = "+i+"  size of tuples = "+tuples.size());
}

System.out.println("Done");
}

public static void SimilarityGenresandRating(ArrayList<tuple> tuples)
{
//Travelling 3 levels for checking the similarity

Queue q = new LinkedList<tuple>();
q.add(tuples.get(0));

int level = 1;
while(!q.isEmpty())
{
tuple t= (tuple) q.remove();
System.out.println("*****************At Level = "+Math.ceil(Math.log(level)));
System.out.println("Male ratio ="+ (t.male*1.0)/(t.male +t.female)+ " Female ratio = "+ (t.female * 1.0)

System.out.println("Genres Present in this cluster = ");
Iterator it = t.genre.entrySet().iterator();

while(it.hasNext())
{
Map.Entry pair = (Map.Entry)it.next();
System.out.println(pair.getKey());
}

System.out.println("Average rating of the users in this age group id = "+t.rating);
System.out.println("Common Profession is = "+t.occupation);
```

```
if(level<8)
{
++level;
if(t.llink != null)
q.add(t.llink);
if(t.rlink != null)
q.add(t.rlink);
}
}
}

public static void main(String[] args)
{
String sql;
    sql = "SELECT m.genre,r.rating,u.age,u.gender,u.occupation FROM users1 u, movies1 m, ratings1 r whe
       "u.userid = r.userid and m.movieid = r.movieid";
    ArrayList<tuple> tuples = new ArrayList<tuple>();

    MySQLConnection mysqlConnection = new MySQLConnection("suhas_db", "root", "root");
    boolean status = mysqlConnection.getData(sql, tuples);

if(status == false)
{
System.out.println("Wrong in data fetch");
System.exit(0);
}
else
{
System.out.println("Total Tuple Strength = "+tuples.size());
}

agglomerative(tuples);

mysqlConnection.stop();

SimilarityGenresandRating(tuples);

}

}
```

Similar to the first clustering algorithm this takes lot of time and memory so restricted to first 1000 ratings.

The output of first three levels are printed below. The output says that there is no common genre that all the people with age group of 28 and profession(15) watch. There is no much similarity with genres of movie they watch. But in gender, this age group has Male ratio of 0.675 and Female ratio of 0.325. The average rating given by these people is 3.788.
The similarity is Male ratio dominates female ratio in the age group of 28 years and their common profession is 15 withing no common genre's of movie watching and average rating of 3.788. One similar thing about gender with all age groups is that male to female ratio is always constant. The average ratings given is also very common in all the clusters formed using genre and rating.

```
/*************************************************************/
```

```
*****************At Level = 0.0
Male ratio =0.675 Female ratio = 0.325
Common Age group is = 28
Genres Present in this cluster =
Average rating of the users in this age group id = 3.788
Common Profession is = 15
*****************At Level = 1.0
Male ratio =0.674 Female ratio = 0.326
Common Age group is = 28
Genres Present in this cluster =
Average rating of the users in this age group id = 3.788
Common Profession is = 16
*****************At Level = 2.0
Male ratio =0.676 Female ratio = 0.324
Common Age group is = 28
Genres Present in this cluster =
Average rating of the users in this age group id = 3.788
Common Profession is = 15
*****************At Level = 2.0
Male ratio =0.672 Female ratio = 0.328
Common Age group is = 29
Genres Present in this cluster =
Average rating of the users in this age group id = 3.788
Common Profession is = 7
*****************At Level = 2.0
Male ratio =0.676 Female ratio = 0.324
Common Age group is = 28
Genres Present in this cluster =
Average rating of the users in this age group id = 3.788
Common Profession is = 16
*****************At Level = 2.0
Male ratio =0.676 Female ratio = 0.324
Common Age group is = 28
Genres Present in this cluster =
Average rating of the users in this age group id = 3.788
Common Profession is = 16
*****************At Level = 2.0
Male ratio =0.676 Female ratio = 0.324
Common Age group is = 28
Genres Present in this cluster =
Average rating of the users in this age group id = 3.788
Common Profession is = 15
*****************At Level = 3.0
Male ratio =0.696969696969697 Female ratio = 0.30303030303030304
Common Age group is = 31
Genres Present in this cluster =
Average rating of the users in this age group id = 3.787878787878788
Common Profession is = 20
*****************At Level = 3.0
Male ratio =0.6630434782608695 Female ratio = 0.33695652173913043
Common Age group is = 29
Genres Present in this cluster =
Average rating of the users in this age group id = 3.7880434782608696
```

```
Common Profession is = 7
*****************At Level = 3.0
Male ratio =0.696969696969697 Female ratio = 0.30303030303030304
Common Age group is = 30
Genres Present in this cluster =
Average rating of the users in this age group id = 3.787878787878788
Common Profession is = 9
*****************At Level = 3.0
Male ratio =0.6684782608695652 Female ratio = 0.33152173913043476
Common Age group is = 28
Genres Present in this cluster =
Average rating of the users in this age group id = 3.7880434782608696
Common Profession is = 16
*****************At Level = 3.0
Male ratio =0.6818181818181818 Female ratio = 0.3181818181818182
Common Age group is = 30
Genres Present in this cluster =
Average rating of the users in this age group id = 3.787878787878788
Common Profession is = 20
*****************At Level = 3.0
Male ratio =0.6739130434782609 Female ratio = 0.32608695652173914
Common Age group is = 28
Genres Present in this cluster =
Average rating of the users in this age group id = 3.7880434782608696
Common Profession is = 16
*****************At Level = 3.0
Male ratio =0.6818181818181818 Female ratio = 0.3181818181818182
Common Age group is = 29
Genres Present in this cluster =
Average rating of the users in this age group id = 3.787878787878788
Common Profession is = 20
*****************At Level = 3.0
Male ratio =0.6739130434782609 Female ratio = 0.32608695652173914
Common Age group is = 28
Genres Present in this cluster =
Average rating of the users in this age group id = 3.7880434782608696
Common Profession is = 15
/*************************************************************/
```

5. *Agglomerative clustering on BigRed2*

Since it was taking lot of time and memory to run agglomerative, I tried running it on IU super-computer BigRed2. However it was not as smooth as I expected. First of all I didn't get rights to install MySQL on BigRed2. So I could no way run the program there. Later I came up with the idea of writing programs to write output of My MySQL into a CSV and upload it to BigRed2 to read it from CSV and do agglomeration. Even though the logic worked I faced problem of no more memory(Even on Super-Computer). I was venturing into submitting jobs to Super-computer but couldn't proceed due to time constraints. I hope I will be able completing it in the next week.

The code for writing the output of MySQL query into CSV is written in **PutDataintotext.java**. The output of the file is present in **agg3.1.txt**. The code for running agglomerative clustering on age and profession on Bigred2 is written in **ReadData.java**. All these files are stored under the folder **BigRed2**.

References

[1] Sergio M Focardi, Frank J Fabozzi, and Turan G Bali. Probability: Random variables and expectations. *Mathematical Methods for Finance: Tools for Asset and Risk Management*, pages 107–145.

[2] Ronald R Hocking. A biometrics invited paper. the analysis and selection of variables in linear regression. *Biometrics*, pages 1–49, 1976.

[3] Deborah J Rumsey. *Statistics for dummies*. John Wiley & Sons, 2011.

[4] Joseph F Hair, William C Black, Barry J Babin, Rolph E Anderson, and Ronald L Tatham. *Multivariate data analysis*, volume 6. Pearson Prentice Hall Upper Saddle River, NJ, 2006.

[5] Pang-Ning Tan, Michael Steinbach, Vipin Kumar, et al. *Introduction to data mining*, volume 1. Pearson Addison Wesley Boston, 2006.

[6] Paul S Bradley, Usama M Fayyad, Cory Reina, et al. Scaling clustering algorithms to large databases. In *KDD*, pages 9–15, 1998.